

Introduction à l'Algorithmique

On utilisera l'éditeur en ligne : Future Coder

A chaque utilisation de la fonction **print** ou **input**, un enfant perd son doudou. Faites attention s'il vous plaît.

I Etude d'un algorithme

On considère l'algorithme écrit en pseudo-code ci-dessous :

Algorithme F

ENTRÉES : n un entier naturel

```
1.  $f \leftarrow 1$ 
2.  $i \leftarrow 0$ 
3. tant que  $i < n$  faire
4.    $i \leftarrow i + 1$ 
5.    $f \leftarrow f * i$ 
6. fin tant que
7. renvoyer  $f$ 
```

Exercice 1. Simulation Papier

On simule l'algorithme F sur l'entrée 5.

1. Combien y a-t-il de tours de boucles? (nombre de fois que l'on exécute le bloc de la boucle).
2. Quelles sont les valeurs de i et f avant chaque exécution de la ligne 3?
3. Donnez une équation vérifiée par i et f avant chaque exécution de la ligne 3?

Exercice 2. Spécification de F

Décrire la sortie de $\mathcal{F}$ sur une entrée quelconque n , entier naturel.

Exercice 3. Vérification

Implémenter l'algorithme sur Future Coder et faire un appel sur l'entrée 5. A l'aide du bouton snoop d'abord, puis du bouton Python Tutor, vérifiez les résultats de votre simulation.

Exercice 4. Somme des entiers

Implémenter une fonction `somme_entiers` qui prend en entrée un entier naturel n et qui renvoie $\sum_{i=0}^n i$.

II Diviser ou ne pas Diviser?

Exercice 5. Divisibilité

Implémenter une fonction `est_diviseur` qui prend en entrée deux entiers naturels a et b et qui renvoie un booléen vrai ssi a est un diviseur de b . (et qui renvoi donc faux si a n'est pas un diviseur de b , quoi d'autre?)

Exercice 6. Premier

Un nombre entier naturel plus grand que 2 est premier ssi il ne possède aucun diviseurs parmi les entiers naturels strictement plus grands que 1 et plus petits ou égaux à sa racine carrée.

Voici une méthode simple, décrite en langue naturelle française, pour savoir si un nombre est premier.

On considère que le nombre, n , est premier, jusqu'à preuve du contraire.

On part de l'entier 2.

Tant qu'on a trouvé un diviseur, ou qu'il reste des nombres entiers à tester entre 2 et la racine carrée de n : on vérifie si l'entier courant est un diviseur de n et si c'est le cas, on note que le nombre n n'est finalement pas premier.

Complétez et implémentez l'algorithme suivant pour qu'il renvoie un booléen vrai ssi n est premier.

L'algorithme essaye de reproduire fidèlement la description précédente. Vous *devrez* utiliser la fonction `est_diviseur`.

Algorithme P

ENTRÉES : n un entier naturel plus grand ou égal à 2

```

1.  $p \leftarrow \dots$ 
2.  $d \leftarrow 2$ 
3. tant que ... faire
4.   si ... alors
5.      $p \leftarrow \dots$ 
6.   fin si
7.    $d \leftarrow d + 1$ 
8. fin tant que
9. renvoyer  $p$ 
```

Simulez l'algorithme sur Python Tutor sur l'entrée 11. Combien d'environnements sont créés au total ? Combien y en a-t-il au plus existants simultanément ?

Exercice 7. Euclide

Implémenter en Python l'algorithme d'Euclide qui prend en entrée deux entiers a et b et qui renvoie le PGCD de ces deux entiers.

https://fr.wikipedia.org/wiki/Algorithme_d%27Euclide

III Suites récurrentes**Exercice 8. Méthode de Halley pour le calcul de $\sqrt{2}$**

Implémenter en Python la fonction `halley` qui calcule le $n^{\text{ème}}$ terme de la suite récurrente suivante :

$$H_0 = 1, H_{n+1} = H_n \cdot \frac{(H_n^2 + 6)}{(3H_n^2 + 2)}$$

Exercice 9. Suite de Fibonacci

Implémentez en Python la fonction `fibonacci` qui calcule le $n^{\text{ème}}$ terme de la suite récurrente d'ordre 2 suivante :

$$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

Exercice 10. Syracuse

Les mathématiques ne sont pas encore prêtes pour de tels problèmes – Paul Erdős

Une suite de Syracuse est définie par la relation de récurrence suivante :

$$S_{n+1} = \begin{cases} \frac{S_n}{2} & \text{si } S_n \text{ est pair,} \\ 3S_n + 1 & \text{si } S_n \text{ est impair.} \end{cases}$$

La conjecture de Syracuse stipule que, quelque soit $S_0 \in \mathbb{N}^*$, la suite de Syracuse débutant par S_0 atteint 1. On notera qu'une fois la valeur 1 atteinte, la suite boucle sur les valeurs 1, 4, 2, 1, ...

Implémentez une fonction `syracuse` qui prend en entrée un entier strictement positif s_0 et qui renvoie le premier rang auquel la suite atteint la valeur 1.

Est-ce que la fonction `syracuse` termine sur toutes les entrées ?

IV Interlude – POUR S'ENTRAÎNER**Exercice 11. Suite géométrique**

Une suite géométrique est une suite définie par récurrence de la forme :

$$u_n = \begin{cases} a & \text{si } n = 0 \\ q * u_{n-1} & \text{sinon} \end{cases}$$

avec $a \in \mathbb{R}$ un terme initial et $q \in \mathbb{R}$ la *raison* de la suite.

Implémenter une fonction `geometrique(a, q, n)` qui renvoie le $n^{\text{ème}}$ terme de la suite géométrique de premier terme a et de raison q .

Exercice 12. Suites adjacentes

Implémentez en Python la fonction `adjacentes` qui prend en argument un flottant ϵ et qui calcule le premier rang n termes des suites adjacentes suivantes, à partir duquel $|a_n - b_n| < \epsilon$:

$$b_{n+1} = \frac{a_n + b_n}{2}, a_{n+1} = \frac{2}{b_{n+1}}, a_0 = 1, b_0 = 2$$

Exercice 13. Méthode de Héron

La suite x^a définie ci-dessous converge vers $\sqrt{a}$.

$$\forall n \in \mathbb{N} \quad x_{n+1}^a = \frac{x_n^a + \frac{a}{x_n^a}}{2}, \quad x_0^a = a$$

Implémentez une fonction `heron` qui prend en argument deux entiers a et n positifs et renvoie x_n^a .

Trouvez le rang à partir duquel x_n^2 approche $\sqrt{2}$ à 10^{-10} près (≈ 1.4142135623).

Exercice 14. Nombre parfait

Un nombre parfait est un nombre entier positif égal à la somme de ses diviseurs stricts (lui-même exclu).

Implémentez une fonction `parfait` qui prend en entrée un entier et renvoie vrai s'il est parfait.

V Interlude – POUR ALLER PLUS LOIN**Exercice 15. Logarithme entier**

On rappelle que pour tout $n \geq 1$ et $b \geq 2$, le logarithme entier en base b de n est l'unique entier l tel que

$$b^l \leq n < b^{l+1}$$

Implémentez en Python la fonction `log_entier` qui prend en entrée un entier $n \geq 1$ et un entier $b \geq 2$ et renvoie le logarithme entier de n en base b .

Exercice 16. Nombres Chanceux d'Euler

Un nombre chanceux d'Euler est un nombre $n > 1$ tel que pour tout $0 \leq i \leq n-2$, $i^2 + i + n$ est premier.

Implémentez une fonction `chanceux_euler` qui prend en entrée un entier n et renvoie vrai ssi n est chanceux.

Exercice 17. Fonction 91 de McCarthy

La fonction 91 de McCarthy est définie pour $n \in \mathbb{N}$ par la formule suivante :

$$f(n) = \begin{cases} n - 10 & \text{si } n > 100 \\ f(f(n + 11)) & \text{sinon.} \end{cases}$$

Implémentez une fonction `mccarthy` qui prend en entrée un nombre entiers n positifs et renvoie $f(n)$. Observer le retour de la fonction pour les valeurs plus petites ou égales à 101.